

IN THE CLAIMS

1 (canceled)

2 (previously presented): A method of launching a software application in a hand-held device, comprising:

receiving an abbreviated textual command in a natural language search engine, entered by a user of the hand-held device; and

while receiving the abbreviated textual command, performing the steps of:

searching a natural language database that stores a data set of abbreviated textual commands and associated application commands;

analyzing historical preferences to determine one or more probable complete commands matching a currently received portion of the abbreviated textual command; and

displaying a list of probable complete commands matching the currently received portion of the abbreviated textual command.

3 (previously presented): The method of claim 2, comprising the additional step of:

if a user selects a complete command from the list, then setting the complete command as the abbreviated textual command, and executing the associated application command.

4 (previously presented): The method of claim 2, comprising the additional step of:

if a user does not select a complete command from the list, then receiving an entire abbreviated textual command in the natural language search engine.

5 (previously presented): The method of claim 4, further comprising:

if the abbreviated textual command has an exact match in the data set, then setting the exact match as a user command;

if the abbreviated textual command does not have an exact match in the data set, then analyzing historical preferences to determine if the abbreviated textual command has a probable match in the data set;

if the abbreviated textual command has a probable match in the data set, then setting the probable match as the user command;

if the abbreviated textual command does not have a probable match in the data set, then presenting a list of possible commands, receiving a command choice, and setting the command choice as the user command; and

executing the user command.

6 (previously presented): The method of claim 2, wherein the step of analyzing historical preferences is performed using a set of probability factors that are generated based on historical preferences, where the abbreviated textual command has a probable match in the data set when a probability factor associated with the probable match is greater than a predetermined value.

7 (previously presented): The method of claim 6, wherein the predetermined value is defined by a user.

8 (previously presented): The method of claim 6, comprising the additional step of:

adjusting the set of probability factors each time the abbreviated textual command is entered into the hand-held device.

9 (previously presented): The method of claim 2, wherein:

the abbreviated textual command has a first component and a second component, wherein the first component represents a desired application command, and the second component represents a desired application tag; and

the natural language database stores a data set of abbreviated textual commands and associated application commands and tags.

10 (previously presented): The method of claim 2, wherein the abbreviated textual command is entered into a graphical dialog box.

11 (previously presented): The method of claim 2, wherein the natural language search engine can receive the abbreviated textual command while any of the software applications are executing.

12 (previously presented): The method of claim 5, wherein the list of possible commands presented if the abbreviated textual command does not have a probable match in the data set includes a set of recently executed application commands.

13 (previously presented): The method of claim 5, wherein the list of possible commands presented if the abbreviated textual command does not have a probable match in the data set includes a set of generic application commands that the natural language search engine is capable of executing.

Claims 14-36 (canceled)

37 (previously presented): A method comprising:

storing a data set of abbreviated textual commands and corresponding complete commands;
receiving a portion of an abbreviated textual command being entered by a user; and
before receiving the entire abbreviated textual command, comparing the received portion of the abbreviated textual command to the stored abbreviated commands to determine a probable subset of the complete commands.

38 (previously presented): The method of claim 37 further comprising after the comparing step:

displaying the probable subset of the complete commands to the user.

39 (previously presented): The method of claim 38 further comprising after the displaying step:

receiving an indication of which of the displayed complete commands a user chooses; and
executing the chosen complete command.

40 (previously presented): The method of claim 38 further comprising after the displaying step:

receiving a further portion of the abbreviated textual command; and
narrowing the probable subset based on the further portion received.

41 (previously presented): The method of claim 37 further comprising:

when the probable subset consists of only one complete command, executing that one complete command.

42 (previously presented): The method of claim 37 wherein the storing step includes a user assigning which complete commands should correspond in the future to which abbreviated textual commands.

43 (previously presented): The method of claim 37 wherein the storing step includes generating the data set based on which abbreviated textual commands a user has historically used for choosing each complete command.

44 (previously presented): The method of claim 37 wherein the comparing step includes:

if the data set indicates that the user has chosen to execute a particular complete command more than a predetermined percentage of the time less than 100% after having entered an abbreviated textual command matching the currently received portion of text, then narrowing the subset to that command.

45 (previously presented): The method of claim 44 wherein the predetermined percentage is 50%.

Claims 46-47 (canceled)

48 (previously presented): A method comprising:

receiving a text string being entered by a user;

while receiving the text string, comparing a received portion of the text string to stored text commands to determine which of the stored text commands is a probable text command based on a portion of the probable text command matching the received text string; and

initiating a software operation corresponding to the probable text command.

49 (previously presented): The method of claim 48 wherein said portion of the probable text command is not the entire text command.

50 (previously presented): The method of claim 48 wherein the comparing step includes:

identifying a plurality of the stored text commands that have portions matching the received text string; and

determining which one of the plurality is the probable text command based on historical preferences.

51 (new): A mobile communication device comprising:

a database of user commands for initiating respective software applications; and
a natural language search engine configured to:

receive a command text string being entered by a user, the text string being in two-part format with one part being an abbreviation for a software application and the other part being an abbreviation for an object of the application;

compare the text string to the user commands in the database;
if, in the comparing step, the text string does not match any of the database's user commands, then:

display a list of generic commands to the user.

receive, from the user, an indication of which of the generic commands the user selects from the list; and

execute the selected generic command by initiating the corresponding application.

52 (new): A mobile communication device comprising:

a database of user commands for initiating respective software applications; and
a natural language search engine configured to:

receive a command text string being entered by a user while the user is in a first application, the text string being in two-part format with one part being an abbreviation for a software application and the other part being an abbreviation for an object of the application;

compare the text string to user commands in the database, to determine which of the user commands matches with the text string; and

execute the matching user command by initiating the corresponding second application;

whereby the user launches the second application from within the first application by entering the command text string within the first application.

53 (new): The method of claim 54 wherein the first application is a email composing application, and the second application is a calendar application.

54 (new): A mobile communication device comprising:

a database of user commands for initiating respective software applications; and
a natural language search engine configured to:

receive a command text string being entered by a user, the text string being in two-part format with one part being an abbreviation for a software application and the other part being an abbreviation for an object of the application;

determine a probability that a particular command is likely desired by the user based both on the number of times the user has historically executed the particular application after entering the received text string and on the amount of time since the command was last invoked; and

execute the particular command based on its probability being above a threshold, despite its probability being less than 1, and without determining the probabilities for other commands that might be desired by the user.

55 (new): The method of claim 56 wherein the threshold is 0.5.

56 (new): A mobile communication device comprising:

a database of user commands for initiating respective software applications; and
a natural language search engine configured to:

receive a command text string being entered by a user; and

display a list of frequently used commands from the database as soon as the user begins entering the command text string, for the user to select one of the commands from the list.